Reusable transformations of Data Cube Vocabulary datasets from the fiscal domain

Jindřich Mynarz, Jakub Klímek, Marek Dudáš, Petr Škoda, Christiane Engels, Fathoni A. Musyaffa, Vojtěch Svátek

SemStats 2016
October 18, 2016

OpenBudgets.eu

Motivation

Common data models, such as the Data Cube Vocabulary, make **common transformations** applicable to a variety of datasets. While data models can be shared as RDF vocabularies, there is no established way of **sharing data transformations**.

Driven by the OpenBudgets.eu project, we proposed a way of modularizing and **reusing data transformations as pipeline fragments** to prevent duplication of effort and speed up the development of ETL pipelines.

OpenBudgets.eu project

http://openbudgets.eu is an open government project funded by the European Union that is focused on developing an open-source **analytical platform for budget and spending data**.

The platform will use a **semantic data model** based on the W3C standard **Data Cube Vocabulary**. The data model will provide leverage for **visualizations** and **comparative analysis tools**.

Use cases in **data journalism**, **transparency**, and **participatory budgeting**.

LinkedPipes ETL

LP-ETL (http://etl.linkedpipes.com) is a data processing tool focused on producing RDF using ETL procedures.

Data processing tasks are defined as **pipelines** combining **configurable components** into data flow graphs. Pipelines are stored in RDF serialized into **JSON-LD**.

Example components:

- **Extract:** HTTP GET, SPARQL endpoint
- **Transform:** Tabular, SPARQL update
- **Load:** Files to SCP, Graph Store Protocol

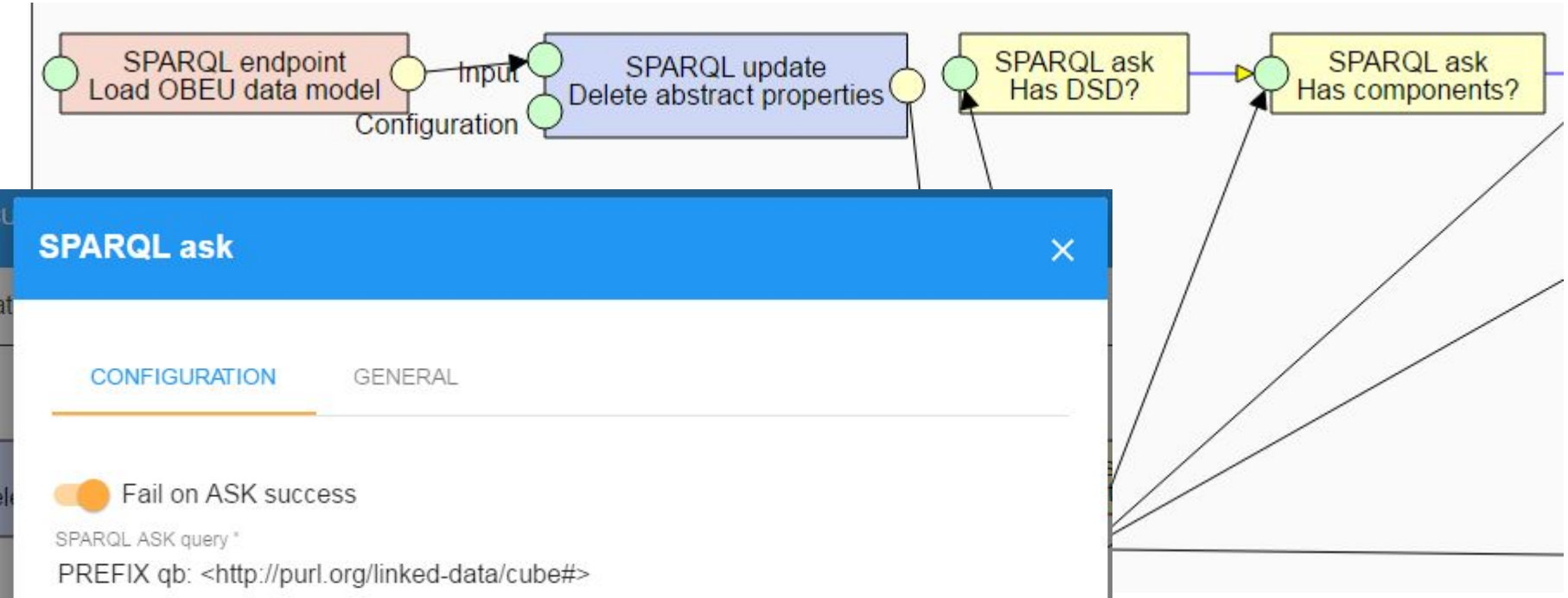
Reusable transformations in LP-ETL

Pipeline fragments: partial pipelines can be exported (without sensitive configuration), shared on the Web, and imported into other pipelines. Contract of the fragment's interface may be checked by SPARQL ASK assertions.

Runtime configuration: component configuration can be generated dynamically based on input data, e.g., using SPARQL CONSTRUCT

Pipeline input via HTTP POST: pipeline execution can be triggered by sending an HTTP POST request; easing integration with other tools.

Component Usage Example: SPARQL ask



SPARQL ask

CONFIGURATION GENERAL

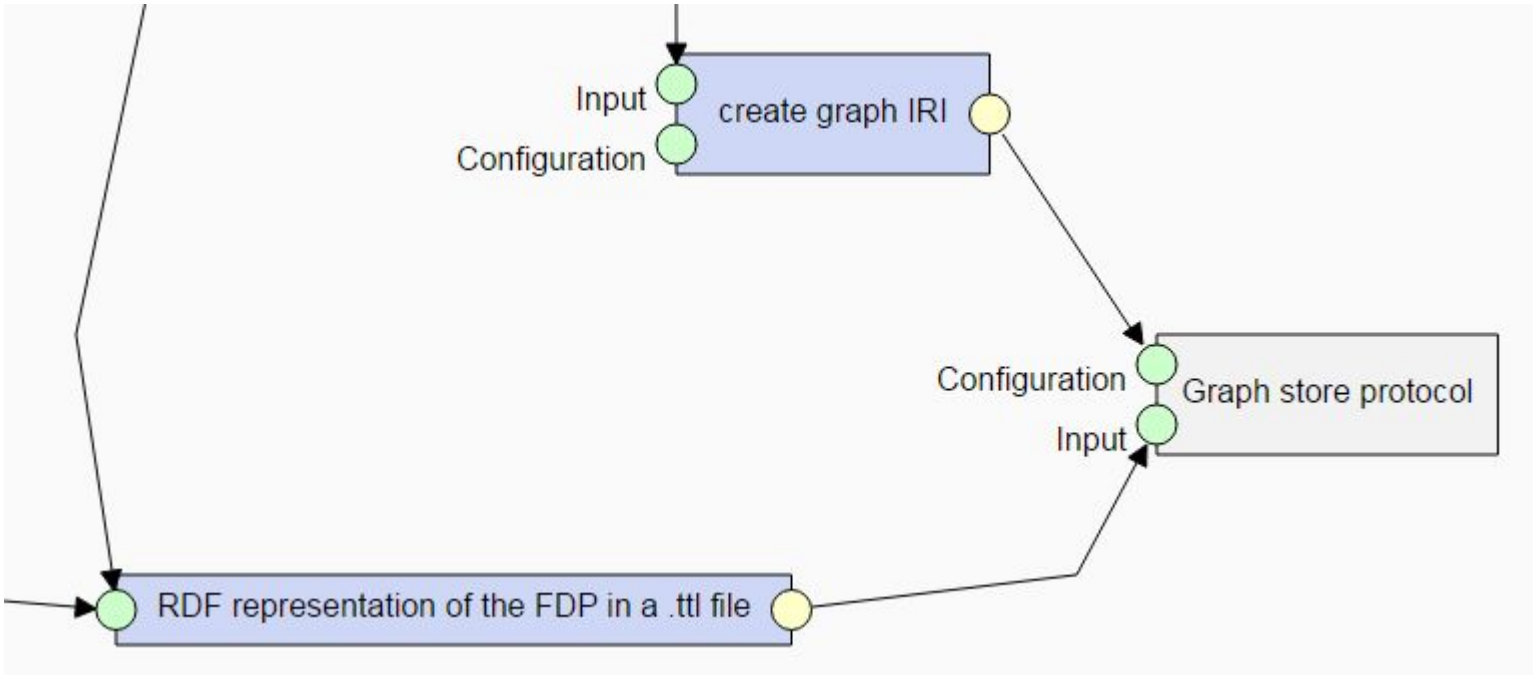
Fail on ASK success

SPARQL ASK query *

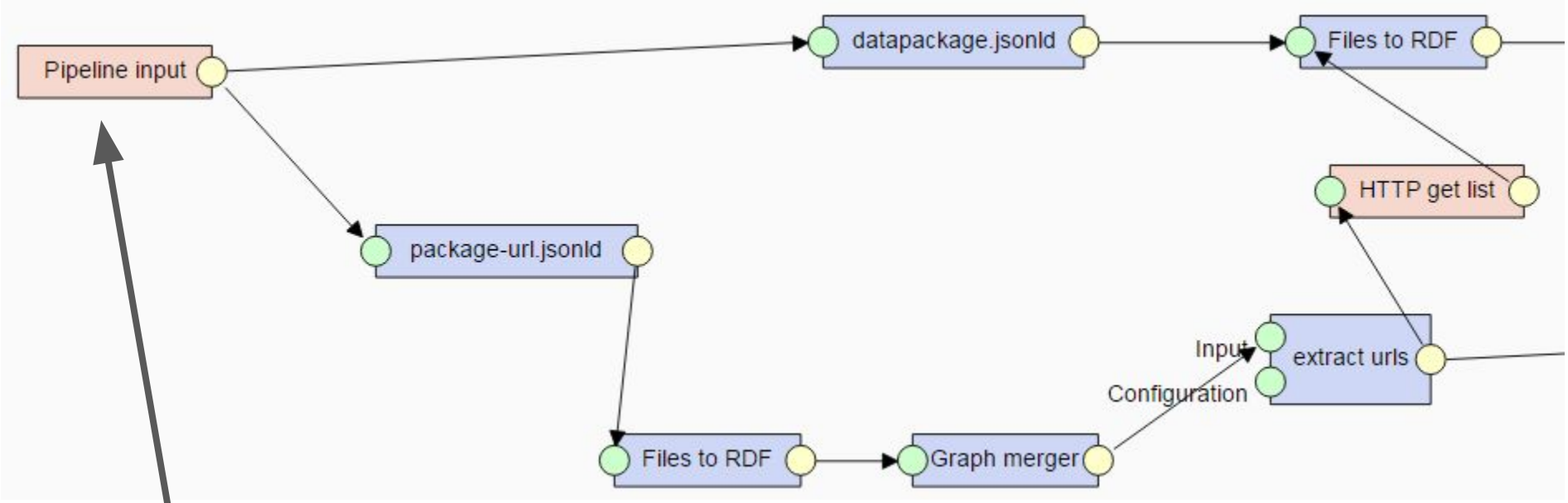
```
PREFIX qb: <http://purl.org/linked-data/cube#>

ASK {
  [] a qb:DataStructureDefinition ;
    qb:component/(qb:componentProperty|qb:dimension|qb:measure|qb:attribute) ?
componentProperty .
  FILTER NOT EXISTS {
    ?componentProperty ?p [] .
  }
}
```

Component Usage Example: Configuration Input

SPARQL construct creates IRI of the target graph for loading RDF into a triple-store.

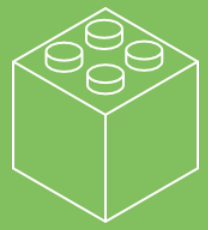


Component Usage Example: Pipeline Input




```
curl -i -X POST -H "Content-Type: multipart/form-data" -F
"input=@datapackage.jsonld"
http://localhost:8181/resources/executions?pipeline=http://localhost:8181/res
ources/pipelines/created-1473847447070
```

Reusable transformations



Pipeline fragments of reusable transformations are published and documented at https://github.com/openbudgets/pipeline-fragments.
There are 2 kinds of transformations:

1. Generic transformations for any DCV-compliant data
2. Specific transformations for the fiscal domain

Transformations of DCV data

- **DCV normalization:** reused SPARQL updates from the DCV specification to normalize attachment of data cube's components
- **DCV validation:** DCV's integrity constraints reformulated and optimized as SPARQL CONSTRUCT queries producing SPIN-RDF data + rendered HTML reports
- **DCV to CSV conversion:** generates a SPARQL SELECT query based on dataset's data structure definition (DSD); reflecting component's type, attachment, and order.

Transformations of OB.eu data

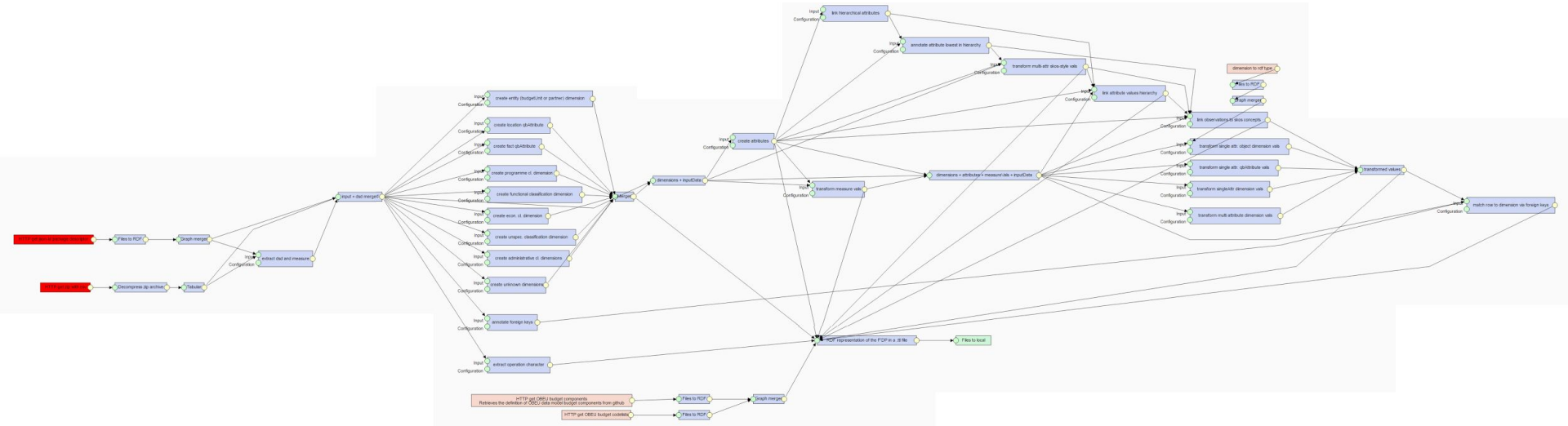
- **Validation of OB.eu integrity constraints:** Similarly to DCV, the OpenBudgets.eu data model defines additional integrity constraints (e.g., missing mandatory property) that are checked using SPARQL CONSTRUCT queries and reported to the user.
- **Normalization of monetary amounts:** Normalization in time and space using external DCV data on **GDP deflators** & **exchange rates** from Eurostat to make monetary amounts better **comparable**.

Transformations of OB.eu data

- **FDP to RDF transformation:**
  Fiscal Data Package (FDP) is a data format of **OpenSpending** based on CSV + JSON descriptor.
- FDP and OB.eu domains are virtually identical
- A series of SPARQL queries transforms the descriptor into OB.eu compliant DSD and the CSV into DCV observations
  - Works but very complicated and slow, reasonable for CSV up to 1MB
  - FDP CSVs can reach hundreds of megabytes
  - We had to reimplement CSV processing in Java - as a new "proprietary" LP-ETL component

FDP to RDF Pipeline

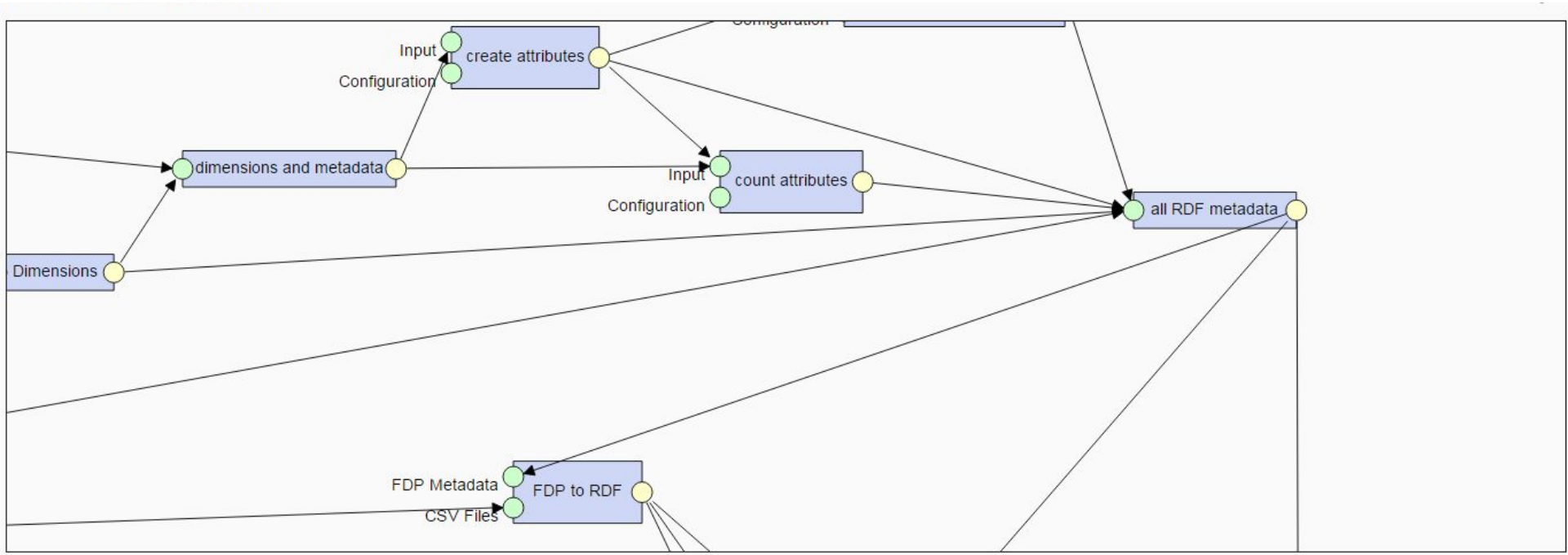
Using SPARQL only - illustrative example of the complexity



FDP to RDF Pipeline

Illustrative example part 2

- CSV to RDF part of transformation now implemented in Java as a new single LP-ETL component
- The descriptor is still processed with SPARQL



OpenBudgets.eu use case

Reuse of the transformations within the OB.eu project:

- **Validation of OpenBudgets.eu datasets:**
  We manually developed ETL pipelines for more than 200 datasets. To **avoid common pitfalls** in this error-prone ETL we used the validation pipelines (DCV normalization + DCV validation + OB.eu validation).
- **Comparative analysis using normalized values:**
  OpenBudgets.eu's scope covers many EU countries, so to improve **comparability** of data for analysis we used the normalization pipeline + DCV to CSV pipeline.

Demo

Conclusions

Common data transformations should be as reusable as **common vocabulary terms**.
We showed how **data transformations can be composed** by importing pipeline fragments via dereferenceable IRIs.
Our future work is to **battle-test the reusable transformations in OpenBudgets.eu** and beyond, for other multidimensional data.

Acknowledgement: The presented research has been supported by the H2020 project no. 645833 (OpenBudgets.eu).